On Fairness as a Rostering Objective

Anna Lavygina
ServicePower
Stockport, UK
a.lavygina@servicepower.com

Kristopher Welsh
Manchester Metropolitan University
Manchester, UK
k.welsh@mmu.ac.uk

Alan Crispin
Manchester Metropolitan University
Manchester, UK
a.crispin@mmu.ac.uk

ABSTRACT

Many combinatorial optimization problems involve scheduling or ordering work that will ultimately be completed by a company's employees. If solution quality is measured by a simple weighted sum of the constraint violations for each employee, an optimizer may produce solutions in which a small number of employees suffer a highly disproportionate share of these violations. We present the results of experiments in generating rosters whilst considering fairness as an additional optimization objective.

CCS CONCEPTS

• **Computing methodologies → Planning and scheduling**; Randomized search; • **Applied computing → Multi-criterion optimization and decision-making**;

KEYWORDS

staff rostering, multiobjective optimisation, constrained optimisation, evolutionary computing, simulated annealing

ACM Reference Format:
Anna Lavygina, Kristopher Welsh, and Alan Crispin. 2019. On Fairness as a Rostering Objective. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3321913

1 INTRODUCTION

Many academic optimization problems model real-world difficulties faced by employers in scheduling and distributing work between their workers. In rostering problems specifically, optimizers seek to schedule workers to cover shifts whilst violating as few constraints as possible and/or at the minimum cost possible.

Constraints in rostering problems may prevent impossible schedules (e.g. two simultaneous shifts), or discourage undesirable scheduling (e.g. shifts without adequate rest between them). The quality of a solution is typically measured by a weighted sum of the constraint violations for each employee. Optimizers that minimise this metric alone can generate rosters where overall quality is achieved at the expense of unfair distribution between workers' individual schedules.

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic
© 2019 Copyright held by the owner/author(s). Publication rights licensed to the Association for Computing Machinery.
ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3321913

In real-world problems, significant disparity in the distribution of constraint violations between employees would be considered unfair, particularly by the employees allocated less desirable schedules. Unfairness in generated rosters has obvious potential to significantly undermine the adoption of any automated rostering system, and would clearly adversely affect employee morale and retention.

In this paper we explore fairness, defined as a deviation of individual workers' schedule constraint violation penalties, as an optimization objective. We use multi-objective optimization (MOO) approaches to generate good quality solutions that are also fair.

2 UK JUNIOR DOCTOR ROSTERING PROBLEM

We added a fairness objective to benchmark problem sets introduced in [3]. These problem sets are doctor rostering problems simulating hospital departments varying in size and complexity, ranging from 12 to 40 doctors. The problems differ in the number of doctors available, the coverage requirements for a range of shifts throughout the day, the number of doctors contracted to work fewer hours per week, and the number and type of working patterns.

When generating rosters for these problems, the shift coverage requirements and adherance to working patterns are considered hard constraints. Soft constraints are used to represent a number of restrictions imposed on doctors' schedules by the UK's national Junior Doctor Contract. Examples of these constraints include a limit on the length of the average working week, a minimum rest period between shifts and a longer rest period after a sequence of night shifts. The objective used in [3] is the minimization of a total penalty for soft constraint violations:

$$totalPenalty = \sum_{i=1}^{N} \sum_{j=1}^{C} penalty_i^j \qquad (1)$$

where N is the number of employees, C is the total number of constraints, and $penalty_i^j$ is a penalty for violating constraint j by the schedule of employee i.

3 CHOICE OF FAIRNESS METRIC

Fairness in rostering problems is often defined as the balance in workload between workers (e.g. total hours, days, or shifts)[6]. However, in [4] fairness is defined as the maximum number of constraint violations in an individual schedule.

Results using this "worst case" fairness metric in MOO experiments were adverse: solutions were no fairer when considering the metric than without its consideration. Moreover, in some instances, fairer solutions were produced when optimizing only for the global minimum penalty. We posit that this is likely because undesirable changes are not penalized unless the individual schedule currently worst in the solution is affected.

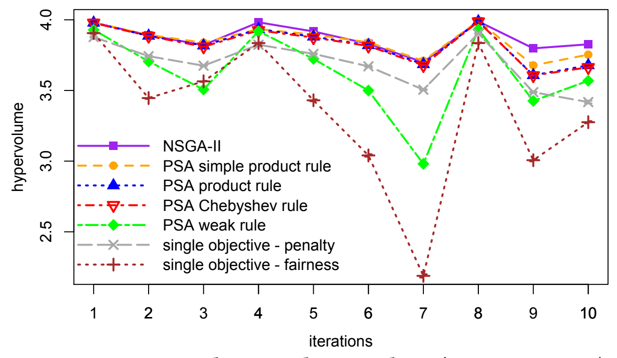

Figure 1: Average hypervolume values (instances 1-10)

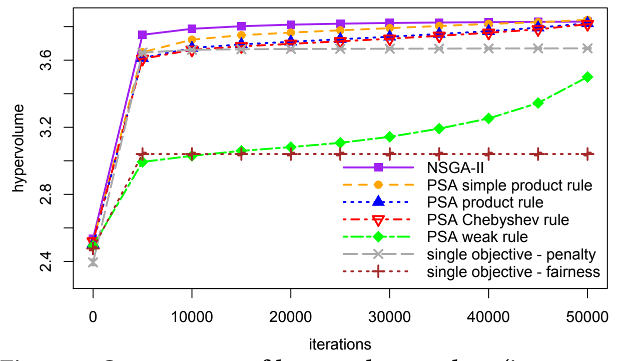

Figure 2: Convergence of hypervolume values (instance 6)

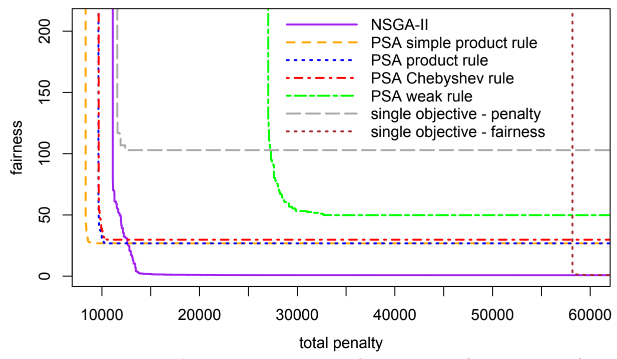

Figure 3: Median attainment functions (instance 6)

We propose a more rigorous definition of fairness, that better captures the distribution in constraint violations in a solution:

$$fairness = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\sum_{j=1}^{C} penalty_i^j - \overline{penalty})^2} \tag{2}$$

where $\overline{penalty} = 1/N \sum_{i=1}^{N} \sum_{j=1}^{C} penalty_i^j$ is the average penalty.

4 ROSTERING APPROACHES

We examine two MOO methods: Pareto Simulated Annealing (PSA) and Non-dominated Sorting Genetic Algorithm II (NSGA-II). The results of these methods are also compared to results obtained by optimising total penalty and the fairness metric as single objectives by a genetic algorithm. The genetic algorithm had the same setup as NSGA-II with the difference that a single objective is considered by the parental and environmental selection operator.

4.1 Pareto Simulated Annealing

We applied the PSA procedure described in [1]. Initial solutions are constructed as described in [3]. In each iteration of the algorithm each solution is modified using a local operator, for example by swapping a shift or series of shifts between employees. Only swaps that do not violate hard constraints are allowed.

If a new solution dominates the old one, the old solution is replaced by the new solution. Otherwise, it may replace the old solution with a certain probability defined by a transition rule. We tested four types of transition rules introduced in [5]: simple product rule, product rule, Chebyshev rule and the weak rule. The set of non-dominated solutions is updated to include new non-dominated solutions.

4.2 NSGA-II

We used a variation of NSGA-II [2] without crossover. On each iteration of the algorithm tournament selection is used to select parent solutions. Offspring solutions are generated by copying parent solutions and applying a local operator as mutation. Parent and offspring solutions are then combined in one set, and sorted based on levels of non-domination and crowding distance. Top 50% solutions are selected for the next iteration.

5 EXPERIMENTAL RESULTS & DISCUSSION

Figure 1 depicts average hypervolume values achieved by different optimization settings across 30 runs for all 10 instances after 50000 iterations (error bars are omitted since they were too small). On almost all instances, NSGA-II performs as well or better than PSA. A significant exception is instance 7, where it is outperformed by PSA with the simple product rule.

NSGA-II has shown to converge quickest to a relatively high hypervolume value on all problem instances as indicated for problem instance 6 in Figure 2. This is due to better fairness metric scores (Figure 3). Conversely, PSA outperformed NSGA-II in terms of total penalty, except in the case where it is used with the week rule. Thus, PSA can be preferable when lower values of total penalty are desirable. Moreover, using MOO helped to achieve better values of total penalty than optimising it as a single objective.

6 ACKNOWLEDGEMENT

This work is supported by the Knowledge Transfer Partnerships scheme (project No.10256).

REFERENCES

[1] Piotr Czyżżak and Adrezej Jaszkiewicz. 1998. Pareto simulated annealing – a metaheuristic technique for multiple-objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis* 7, 1 (1998), 34–47.
[2] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (2002), 182–197.
[3] Anna Lavygina, Kris Welsh, and Alan Crispin. 2017. Doctor rostering in compliance with the new UK junior doctor contract. In *International Conference on Combinatorial Optimization and Applications*. Springer, 394–408.
[4] Djamila Ouelhadj, Simon Martin, Pieter Smet, Ender Ozcan, and G Vanden Berghe. 2012. Fairness in nurse rostering. (2012).
[5] Paolo Serafini. 1994. Simulated annealing for multi objective optimization problems. In *Multiple Criteria Decision Making*. Springer, 283–292.
[6] Jorne Van den Bergh, Jeroen Beliën, Philippe De Bruecker, Erik Demeulemeester, and Liesje De Boeck. 2013. Personnel scheduling: A literature review. *European Journal of Operational Research* 226, 3 (2013), 367–385.